Early Experiences in Traffic Engineering Exploiting Path Diversity: A Practical Approach

Luca Muscariello, Diego Perino

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Early Experiences in Traffic Engineering Exploiting Path Diversity: A Practical Approach

Luca Muscariello — Diego Perino

N° 6474

February 2008

Thème COM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6474--FR+ENG

Early Experiences in Traffic Engineering Exploiting Path Diversity: A Practical Approach

Luca Muscariello * , Diego Perino†

Thème COM — Systèmes communicants
Projets Gang

Rapport de recherche n° 6474 — February 2008 — 21 pages

Abstract: Recent literature has proved that stable dynamic routing algorithms have solid theoretical foundation that makes them suitable to be implemented in a real protocol, and used in practice in many different operational network contexts. Such algorithms inherit much of the properties of congestion controllers implementing one of the possible combination of AQM/ECN schemes at nodes and flow control at sources.

In this paper we propose a linear program formulation of the multi-commodity flow problem with congestion control, under max-min fairness, comprising demands with or without exogenous peak rates. Our evaluations of the gain, using path diversity, in scenarios as intra-domain traffic engineering and wireless mesh networks encourages real implementations, especially in presence of hot spots demands and non uniform traffic matrices.

We propose a flow aware perspective of the subject by using a natural multi-path extension to current congestion controllers and show its performance with respect to current proposals. Since flow aware architectures exploiting path diversity are feasible, scalable, robust and nearly optimal in presence of flows with exogenous peak rates, we claim that our solution rethinked in the context of realistic traffic assumptions performs as better as an optimal approach with all the additional benefits of the flow aware paradigm.

Key-words: Multi-path Routing, Congestion control, Traffic Engineering, Flow-Aware Architectures

* luca.muscariello@orange-ftgroup.com
† diego.perino@orange-ftgroup.com

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Ingènierie du Traffic par Flot
Une Approche Multi-chemins Pratique

Résumé : Les algorithmes de routage dynamique possèdent de solides fondements théoriques qui les rendent aptes á une implémentation rèelle dans diffèrents rèseaux opèrationnels. Ces algorithmes possèdent de nombreuses propriètès propres aux contrôles de congestion grâce á l'utilisation de mécanismes de signalisation explicite et de contrôle des flots á la source.

Dans cet article, nous proposons une formulation linèaire du problème de multiflot avec contrôle de congestion et critère d'èquitè de type "max-min". Les performances obtenues par l'exploitation de chemins multiples sont encourageants, que cela soit en routage intra-domaine ou dans les rèseaux mesh sans fil, et incitent á une implèmentation rèelle, en particulier dans le cas de matrices de trafic non-uniformes et de "points chauds" de demandes.

Nous proposons une approche par flots qui intègre naturellement les multi-chemins aux mècanismes actuels de contrôle de congestion, et nous l'èvaluons par rapport aux solutions actuelles. Les architectures par flot sont rèalisables, robustes, capables de passer á l'èchelle, et quasi-optimales lorsque les flots ont un dèbit-crête explicite. Dans un contexte rèaliste, notre solution possède donc les propriètès d'une solution optimale ainsi que les avantages de l'approche par flots.

Mots-clés : Multi-chemins, Contrôle de congestion, Ingènierie du traffic, Architecture par flots

1 Introduction

Traffic engineering is usually perceived as an off-line functionality in order to improve performance by better matching network resources to traffic demands. Optimisation is performed off-line as demands are averages over the long term that do not consider traffic fluctuation over smaller time scales. Intra-domain routing optimisation, by means of OSPF cost parametrisation [8, 9], is typical example of the aforementioned problem. The max degree of responsiveness is guaranteed at the long term, in a daily or weekly basis for instance. The objective of ISPs is to get rid of a given traffic matrix at the minimum cost, which is estimated as global expenditures for link upgrades. Therefore minimising the maximum link load is a natural objective. Such an engineered network is not robust to any finer grained traffic fluctuation, as large number of failures, traffic flash crowds, BGP re-routes, application re-routing. In current backbones these effects are mitigated by over-provisioned links. In other contexts, of increasing importance nowadays as wireless mesh networks, scarceness of resources push network engineers to accept a higher degree of responsiveness in order to exploit any single piece of unused capacity. We only consider wireless mesh resulting from a radio engineered network, where links guarantee a minimum availability.

Literature on optimal routing in networks starts from seminal works as [3, 4] or [6, 10]. This comprises centralised and decentralised strategies proposed in the very beginning of the ARPAnet project. Historically the main outcome of optimal routing has been shortest-path or, at most, minimum-cost routing. Early experiences on dynamic routing [37] have kept it back in spite of its potential, as sensitivity to congestion has long been refused for being complex, unstable, not prone to be easily deployed.

There is a quite large recent literature on multi-path routing, raised from the need to develop dynamic yet stable algorithms, sensitive to congestion for many applications that could effectively exploit unused resources within the network.

In the Internet, we believe that very specific applications can tolerate such dynamic environment. Adaptive video streaming is, probably, the main application that would definitely be able to tolerate variability and, at the same time, take advantage of unused capacity. It is worth recalling that video applications are going to be the main part of Internet traffic very soon. Video streams usually last very long, and will probably last longer as better applications and contents will be available. Furthermore it is likely that such contents will be transported by a number of non-specified protocols, subject to non better specified fairness criteria.

Conversational applications should be kept away from being routed over multiple routes but also other data services that are made of flows that last potentially very short. Also, much of the present data traffic if it includes mail, web, instant messaging. However, P2P file sharing applications or content delivery networks (CDN) are prone to well exploit path diversity as they intrinsically are robust to rate fluctuations. Other data applications are downloads of software updates even though they can be included in the class of P2P file sharing systems.

In the last few years, intense research on multi-path routing has progressed. Theory of optimisation has been applied to develop distributed algorithms solving a global optimisation problem, see [11, 12, 13, 14, 21, 31, 35, 38]. Optimisation explicits the problem of resource allocation under a chosen fairness criteria. Control theory has been used to obtain delay stability of distributed optimal schemes, see [11, 16, 21, 35]. Other research considers dynamic flow level models [22, 23, 24, 30] in order to take into account arrivals and departures of user's sessions.

In this paper we support the deployment of a flow aware architecture exploiting path diversity for a specific class of applications, rate adaptive video streaming for instance or CDN and P2P file sharing. In such set up we show that flow aware paradigm is nearly optimal for any tipical traffic demand requiring the use of multiple paths without the need to assume any kind of common transport protocol among users, any common fairness semantic and any kind of cooperation between users, and network nodes as well.

In Section 2 we explain our definition of network flows while in Section 3 the modelling framework for optimal routing and congestion control is introduced. The section includes a set of examples on toy networks.

Finally we introduce our main outcome in term of performance evaluation of the optimal solution of large problems, with an original linear program formulation under max-min fairness that is used to evaluate large problems in Section 4.

Section 5 introduces our main original outcome as a new multi-path congestion controller called MIRTO. The algorithm is born inferring an optimal strategy from previous sections. Moreover this algorithm is evaluated within the framework of a flow aware architecture, bringing new arguments in favour of such network paradigm.

2 Traffic Characteristics

IP traffic on a network link can be considered as a superposition of independent sessions, each session relating to some piece of user activity and being manifested by the transmission of a collection of flows.

Sessions and flows are defined locally at a considered network interface. Flows can generally be identified by common values in packet header fields (e.g., the 5-tuple of IP addresses, port numbers and transport protocol) and the fact that the interval between such packets is less than some time out value (20s, say). It is not usually possible to identify sessions just from data in packets and this notion cannot therefore be used for resource allocation. Nevertheless we are more inclined to think about user sessions than protocol defined flows.

A more significant flow characteristic is the exogenous peak rate at which a flow can be emitted. This is the highest rate the flow would attain if the link were of unlimited capacity. This limit may be due to the user access capacity, the maximum TCP receive window, or the current available bandwidth on other links of the path, or the stream rate in case of video applications for instance.

In the rest of the paper we will use interchangeably the terms demand and flow.

3 Modelling framework

The network topology is modelled by a connected graph $G = (N, L)$ given as a set of nodes and links. Let $A = [a_{ij}]$ the adjacency matrix, $a_{ij} = 1$ if there exists a directional link between i and j and $a_{ij} = 0$ otherwise.

The network carries traffic generated by a set of demands Γ, each demand d is given with a triple (s^d, e^d, p^d), with $s \in \mathcal{S}$, $e \in \mathcal{E}$, source and destination nodes with $\mathcal{S}, \mathcal{E} \subseteq N$, and $p \in \mathbb{R}^+$ exogenous peak rate. In our model a network flow d gets a share x_{ij}^d of the capacity c_{ij} at each link $0 \leq x_{ij}^d \leq \min(c_{ij}, p)$. $x_{ij}^d(t)$ is a fluid approximation of the rate at which the source d is sending at time t through link ij.

The network flow can be slit among different paths that are made available by a network protocol at an ingress node. We make no modelling assumption whether paths are disjoint, however the ability to create more path diversity helps design highly robust network routing protocols.

In paragraph 3.3 we model route selection and bandwidth sharing as an optimisation problem that maximises user satisfaction and minimise network congestion under a specified fairness criteria.

3.1 Minimum cost routing

The ability to create the set of optimal paths at the ingress of the network and make them available to the routing protocol requires a certain knowledge of the network status, as link load, path delay and length. However, as this can be done in practice by disseminating local measures, the protocol must be also robust to state inaccuracy. Assuming perfect knowledge of network state, optimal routing can be formulated through the following non linear optimisation problem

Symbol	Meaning
N	node set
L	link set
Γ	demand set
d	demand number
$\mathcal{S}$	source set
$\mathcal{E}$	destination set
s^d	demand d source node
e^d	demand d destination node
p^d	demand d exogenous peak rate
P^d	path set of demand d
k	path number
L_k^d	link set of demand d over its k^{th} path
C_{ij}	capacity of link (i, j)
x_{ij}^d	rate of demand d over link (i, j)
x_k^d	rate of demand d over its k^{th} path
x^d	rate of demand d $(\sum_k x_k^d)$
ρ_{ij}	load on link (i, j) $(\sum_{d\in\Gamma} \frac{x_{ij}^d}{C_{ij}})$

Table 1: Summary of notation used

with linear constrained.

$$\text{minimise} \sum_{i,j\in N} C\left(\frac{\sum_{d\in\Gamma} x_{ij}^d}{c_{ij}}\right)$$

subject to

$$\sum_{k\in N} a_{ik}x_{ki}^d - \sum_{j\in N} a_{ji}x_{ij}^d = \begin{cases} p_i^d & \text{if } \ i \in \mathcal{S} \\ -p_i^d & \text{if } \ i \in \mathcal{E} \quad \forall d \in \Gamma \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

constraint (4) models zero net flow for relay nodes, positive for source nodes and negative for destination nodes. This allows to obtain optimal routes directly from the optimisation problem. C can be thought modelling the link delay often used in traffic engineering formulations of the multi-commodity flow problem,

$$C(x_{ij}) = \frac{x_{ij}}{c_{ij} - x_{ij}} \tag{2}$$

with this formula the cost function becomes the average delay in a M/M/1 queue as a consequence of the Kleinrock independence approximation and Jackson's Theorem. This problem formulation dates back to [6, 10] in the context of minimum delay routing. Using standard techniques in convex constrained optimisation (convex optimisation over a simplex) in [3, 4], it is shown that the optimal solution always select paths with minimum (and equal) firs cost derivatives for any strictly convex cost function. Therefore the problem can be re-formulated as a shortest path problem where path lengths are the first derivatives of the cost function along the path, that can be written with abuse of notation,

$$C^{'}(x_p^d) = C^{'}\left(\frac{\sum_{d\in\Gamma} x_{ij}^d}{c_{ij}}\right) \tag{3}$$

where x_p^d is the portion of flow of demand d flowing through path p. This is what, in [4], Bertsekas and Gallager call first derivative path lengths. Therefore, at optimum all paths have equal lengths. This fact will be use in the following section repeatedly.

Another plausible objective is to minimise the most loaded link, frequent in traffic engineering network operator's backbone optimization in conjunction with link capacity over-provisioning. In the context of multi-path routing this has been used to design TEXCP [16].

3.2 Bandwidth sharing and fairness

Bandwidth is shared between flows according to a certain objective realised by one transport protocol as TCP for data transfers making use of one of its congestion control protocols (Reno, Vegas, Cubic, high speed etc.) or TCP friendly rate control (TFRC) for adaptive streaming applications. In general these protocols realise different fairness criteria, whilst in this context we assume that all flows are subject to a common fairness objective. The problem formulation dates back to Kelly [18] where this problem is formulated as a non linear optimization problem with linear constrained with objective given by a utility function $U(x)$ of the flow rate x.

$$\text{maximise} \sum_{i\in\mathcal{S},d\in\Gamma} U_d(\phi_i^d)$$

subject to

$$\sum_{k\in N} a_{ik}x_{ki}^d - \sum_{j\in N} a_{ji}x_{ij}^d = \begin{cases} \phi_i^d & \text{if } i\in\mathcal{S} \\ -\phi_i^d & \text{if } i\in\mathcal{E} \quad \forall d\in\Gamma \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\sum_{d\in\Gamma} x_{ij}^d \leq c_{ij} \forall i,j\in L \tag{5}$$

demands are assumed elastic, meaning that they could get as much bandwidth as network status allows, and have no exogenous peak rates. A general class of utility functions has been introduces in [27],

$$U_d(x) = \begin{cases} w_d \log x & \alpha = 1 \\ w_d(1-\alpha)^{-1}x^{1-\alpha} & \alpha\neq 1 \end{cases} \tag{6}$$

if $\alpha\to\infty$ fairness criteria is max-min.

This formulation is widely and succesfully used in network modeling. [11, 12, 19, 20, 21, 33] have considered the problem of single and multiple path routing and congestion control under this framework as constraint (4) may count either a single or a multiple set of routes.

3.3 User utility and network cost

User utility and network cost are two conflicting objective in a mathematical formulation. [12, 13, 16] have used the cost function as TE objective likely modeled by the ISPs in ordered to keep low link loads, i.e. minimise costs for upgrades. [11, 14, 21, 31, 35, 38] have just used the network cost as penalty function in place of hard constraints in the optimisation framework.

Congestion sensitive multiple routes selection can be formulated as a mathematical program with non linear objective and linear constraints.

$$\text{maximise} \sum_{i\in\mathcal{S},d\in\Gamma} U_d(\phi_i^d) - \sum_{i,j\in N} C\left(\frac{\sum_{d\in\Gamma} x_{ij}^d}{c_{ij}}\right)$$

subject to

$$\sum_{k\in N} a_{ik}x_{ki}^d - \sum_{j\in N} a_{ji}x_{ij}^d = \begin{cases} \phi_i^d & \text{if } i\in\mathcal{S} \\ -\phi_i^d & \text{if } i\in\mathcal{E} \quad \forall d\in\Gamma \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$\sum_{d \in \Gamma} x_{ij}^{d} \leq c_{ij} \forall i,j \in L \tag{8}$$

$$\phi_i^d \leq p_d \quad \forall d \in \Gamma \quad \forall i \in \mathcal{S} \tag{9}$$

As a new additional constraints we add exgenous rates as this has significant impact in the process of selection of optimal routes.

In this formulation the user utility is a function of the total flow rate traversing the network through the available paths. In a path-demand formulation the flow rateϕ of a given user can be re-written as the sum of the rates over the set of available paths $\mathcal{P}$, i.e. $\phi = \sum_{p \in \mathcal{P}} \phi_p$. A user is free to coordinate sending rates over the paths jointly, aiming at maximise its own utility. Consider now the folllowing relaxation of the objective

$$U(\sum_{p \in \mathcal{P}} \phi_p) \geq \sum_{p \in \mathcal{P}} U(\phi_p) \tag{10}$$

each user's path would be seen as independent, in other words as if it were a separate user and the fairness objective would be at a path, and not user base. A protocol designed observing such rule would break path coordination, while a network imposing per link fair bandwith sharing, would realise this objective for any multi-path controller regardless its original design.

3.4 Toy Examples

In this section we consider two simple network topologies, a triangle and a square as depicted in fig.3.4 with all available paths. All links are bi-derctional with the same capacity C. Capacity $C_{12} = C_{31}$ is increased from C to $15 \times C$. For both scenarios nodes 1 and 3 send data to a single destination node number 2. We find the global optimum of problem (7), using utility function (6) with $\alpha = 2$ and cost function (2). We assume demands fully elastic.

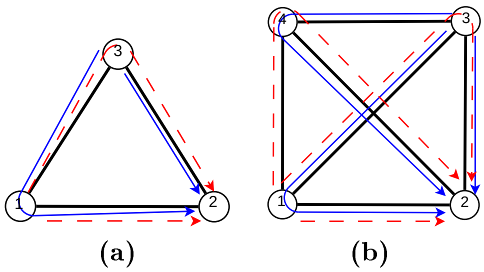

Figure 1: Full mesh triangle and square topologies. Hot-spot destination in node 2 and sources in node 1 and 3.

3.4.1 Triangle

Fig.2 shows the split ratios over the two routes (one-hop and two-hops) and the ratio of the total users' rate (global goodput) over the total consumed network bandwidth (network utilisation). Let us call this ratio GCR (goodput to cost ratio). Top plot relates to coordinated multi-path (CM), whilst bottom plot to uncoordinated (UM). When $C_{12} = C_{31} = C$ the two problems have completely different solutions as CM splits all traffic to the shortest-path and UM splits rates equally. At this stage GCR for CM is 30% larger than UM's. As $C_{12} = C_{31} > C$ increases CM looks for more resources for the second demand along the two-hop path, resulting in more network cost and GCR decreases. However using UM, GCR is insensitive to the split ratios. After a certain point CM and UM have the same performance.

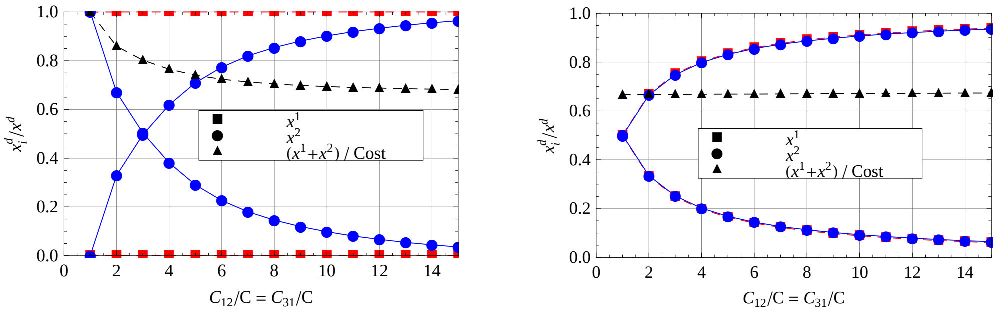

Figure 2: Rate distribution among the available paths and user rate to network cost ratio.

3.4.2 Square

In fig.3 we have similar performance to the triangle, despite CM needs to use more paths to attain the optimum, even to gather a small amount of bandwidth. Furthermore GCR for CM is not that much larger than in case UM is used. In a real protocol, secondary paths would not be used if the attained gain does not meet the cost for the overhead that is not considered here in the model, however significative in practice.

3.4.3 Discussion

MC has a larger stability region, as it consumes less bandwith to provide the same global goodput as UM. This might turn out not be true in practice as weakly used secondary paths could cost to much in terms of over-head due to signalling to set up the connection, or probes to monitor paths that are being coordinated.

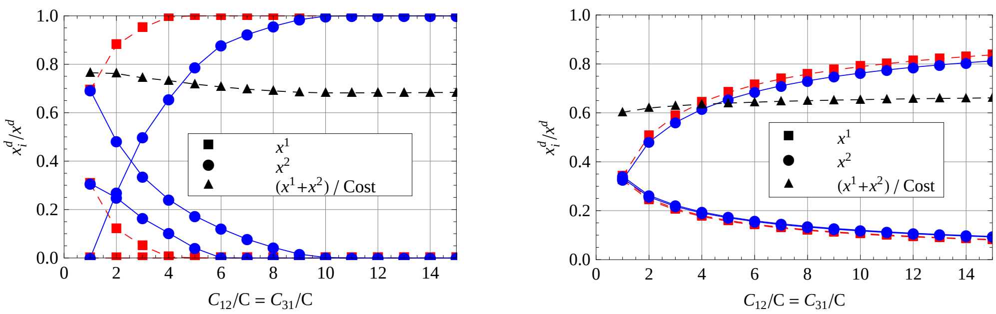

Figure 3: Rate distribution among the available paths and user rate to network cost ratio.

3.5 A Linear Program formulation

In this section we consider problem (7) in sec.3.3 and select one particular fairness criteria: max-min. We write an original formulation of problem (4) as an iterative linear program assuming linear costs $C(x)$. At each iteration a linear sub-problem is solved whom, at optimum, gives the same network flow share to every demand. This share is the maximum bandwidth that can be allocated to the most constrained demand. The network graph G is reduced to $\tilde{G}$ through the following transformation: $\tilde{c}_{ij} = c_{ij} - \sum_{d\in\Gamma} x_{ij}^d$, i.e. capacities are replaced by residual capacities after the allocation of this share of bandwidth. if $\tilde{c}_{ij} = 0$ the link is removed from the graph. The sub-problems are formalised as follows.

$$\text{maximise} \quad z - \sum_{i,j \in N} C\left(\frac{\sum_{d \in \Gamma} x_{ij}^d}{c_{ij}}\right)$$

subject to

$$\sum_{k \in N} a_{ik} x_{ki}^d - \sum_{j \in N} a_{ji} x_{ij}^d = \begin{cases} z & \text{if } \ i \in \mathcal{S} \\ -z & \text{if } \ i \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad \forall d \in \Gamma \tag{11}$$

$$\sum_{d \in \Gamma} x_{ij}^d \leq \tilde{c}_{ij} \forall i, j \in L \tag{12}$$

$$z \leq p_d \quad \forall d \in \Gamma \quad \forall i \in \mathcal{S} \tag{13}$$

In the following section we use this iterative LP in order to obtain the gain that can be obtained exploiting path diversity for large networks with a large number of demands.

4 Analysis of large problems

4.1 Simulation set-up

We study two topologies as shown in fig. 4. The first is the Abilene backbone network [1], while the second is a possible wireless mesh network backhaul. The link capacity distribution is a Normal distribution with average $\bar{C}$ and standard deviation $\bar{C}/10$. The Abilene topology counts $N = 11$ nodes and we perform simulations with mean link capacity set to two scenarios: $\bar{C} = 100Mb/s$ and $\bar{C} = 50Mb/s$. The wireless mesh topology counts $N = 16$ nodes and, similarly, two scenarios are considered: $\bar{C} = 50Mb/s$ $\bar{C} = 25Mb/s$. In this latter case capacities are reduced to represent radio channels with lower available bitrate.

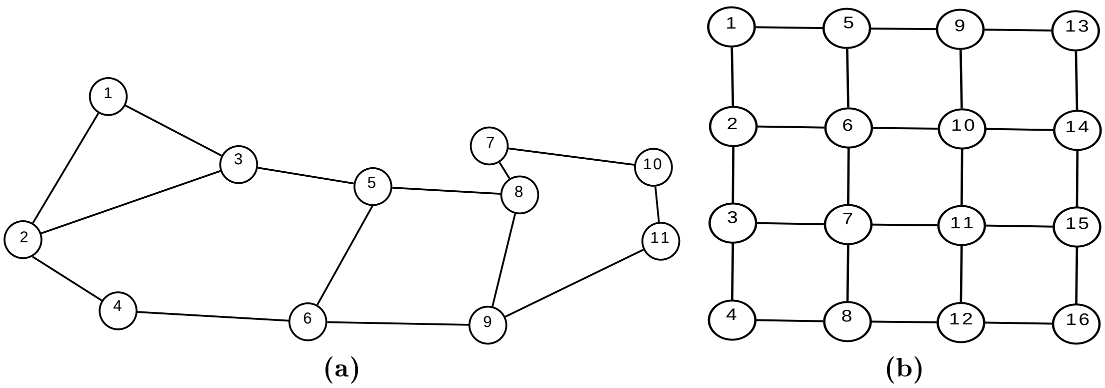

Figure 4: (a) Abilene backbone topology; (b) a planned wireless mesh network.

The distribution of peak rate is taken Log-Normal with parameters $\mu = 16.6$ and $\sigma = 1.04$. These values are taken from a set of fits performed on measurements gathered from Sprint backbone [29] . Two traffic matrices are considered:

- **Uniform**. Every node sends traffic to all other nodes. There are thus $N(N-1)$ demands where N is the number of nodes. To match flow rates to source-destination pairs we use a technique describe in [29], assuming minimum cost-path the routing used. This technique achieves the best match between a set of demands and a set of SD pairs connected by a given routing. This traffic pattern might be representative of an Intra-domain optimised backbone. Since the Abilene network has 11 nodes, in our simulations we generate a total number of 110 demands. We do not consider this traffic matrix over a wireless mesh topology as unlikely all nodes send traffic to all nodes in such networks.

- **Hot-Spot** $|S|$ nodes have r flows directed to a common sink node. We fix the sink node and randomly select $|S|$ source nodes. $|S|r$ demands are randomly assigned to these $|S|$ Source-Sink pairs. In our experiences we set $|S| = 4$, $r = 25$, for a total number of 100 demands. Node 6 is selected as sink in both topologies. This traffic pattern can be rapresentative of a data center located in a backbone topology or a gateway node in a wireless mesh network.

According to the aforementioned set-up we simulate a traffic matrix which is used as input to the LP described in sec.3.5 and solved using the MATLAB optimisation toolbox. For every scenario we evaluate the satisfaction of each demand as the ratio between its attained rate and its exogenous peak rate. A demand is fully elastic if its exogenous rate is larger that the maximum attainable bandwidth in an empty network. Hence, satisfaction is always defined as we need not assume infinite peak rate to elastic flows. We use this performance parameter as it is able to explicit how bandwidth is allocated with respect to the distribution of the exogenous rates. Output data are averaged over multiple runs.

4.2 Numerical results

Results are reported in fig. 5 and 6 and, as expected, show multi-path routing outperforms minimum cost routing over both network topologies and for both traffic patterns. However, the point is to measure the entity of the improvement. In particular, the gain is larger for wireless mesh topology and for scenarios adopting larger link capacities.

The gain of multi-path is, in great part, limited to flows with larger peak rate, whilst flows with lower peak rate are completely satisfied by both routing schemes. This because MinCost routing, under max-min fairness, penalises larger flows by fairly sharing the capacity of the link that acts as bottleneck among all flows there in progress.

Multi-path routing, under max-min fairness, acts similarly except that flows can retrieve bandwidth, not only on their minimum cost path, but also on their secondary routes. Indeed, simulations show that low rate flows do not take any advantage of path diversity, while high rate flows retrieve additional bandwidth along other paths.

Max-min multi-path routing avoids to use more than one path to route low rate flows. This has beneficial effect in practice as it avoids wastage of resources due to the overhead, that might be justified only above a certain minimum rate.

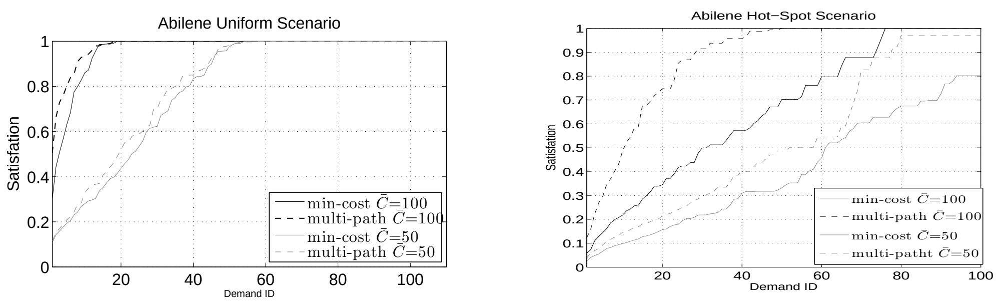

Figure 5: Abilene topology. Satisfaction distribution for uniform (top) and hot-spot (bottom) traffic matrix.

Fig. 5 reports results for the Abilene network. The gain of max-min multi-path routing is very large for hot-spot traffic matrices, and even more significant when network capacities are larger. The routing scheme is able to transfer 65% additional traffic when $\bar{C} = 100$ and 41% when $\bar{C} = 50$, with respect to minimum cost routing. However, under uniform traffic, the gain is much smaller; i.e. only 4% when $\bar{C} = 100$ and 3% when $\bar{C} = 50$.

Under uniform traffic, demands are spread all around nodes and even high rate flows do not exploit path diversity as they are almost completely satisfied over their minimum cost path.

If resources become scarce high rate flows cannot exploit path diversity because all links are already saturated by flows routed along their minimum cost paths. Therefore as the capacity is increased, the gain is distributed to low rate flows first, and to those with higher rate at last.

Of course, as the capacity is fairly large, resulting in a lightly loaded network, both schemes have similar performance as all flows are satisfied along the mincost route.

The variance of the satisfaction is quite small, and between 0.05 and 2.3 for all scenarios. However it is not uniformly distributed among all flows. In fact, large flows are affected by larger variance than small flows. This is due to the max-min fairness criteria that allocates a minimum amount of bandwidth to all flows. This always assures complete satisfaction for small flows while satisfaction of large flows depends on the network capacities that varies from simulation to simulation.

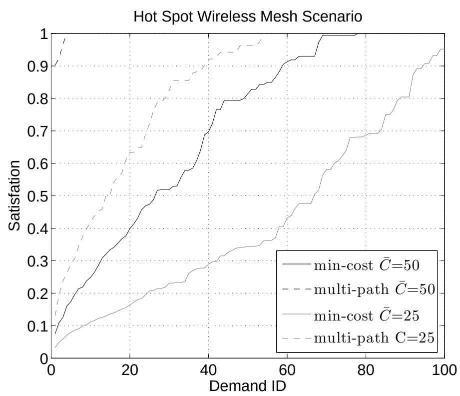

Figure 6: Wireless mesh topology. Satisfaction distribution hot-spot traffic matrix.

5 Architectures and protocols

5.1 Architectures

Any of the proposals for congestion control, and congestion control exploiting path diversity, can be restated in the context of an optimisation problem of the kind of (7) or the uncoordinated counterpart (10). Such problems lead to different architectures based on different theoretical foundations.

We divide such architectures in three groups: fully decentralised, quasi decentralised, flow aware. This classification is for the ease of exposition and for sake of clarity, however it lays itself open to critics.

- **Fully decentralised (FD)**. In such architecture network nodes store, switch and forward packets from an input to an output interface. Neither scheduling nor active queue managament is implemented into the nodes. A form of cooperation is assumed between sources, that implement a common rate control algorithm and, if needed, a common multiple path splitter. In general sources need to be conformant to a common fairness criteria. This is the case for the current Internet and multi-path TCP [11] is one example of such controller.

- **Quasi decentralised (QED)**. Nodes implement any form of active queue management (AQM), to prevent congestion, and explicit congestion notification (ECN) that is triggered according to some temporisation. ECN might also be result of a local node calculation. Sources exploit such notification to control sending rate and split decisions. [12, 13, 16] are two examples in the context of intra-domain traffic engineering.
- **Flow aware (FA)**. Nodes implement packet schedulers that realise fair bandwidth sharing between flows. Sources balance traffic among the available routes subject to the restriction imposed by packet schedulers and autonomously decide how to exploit as better as they can network resources. Congestion control and bandwidth allocation are solved separately. [30] propose a routing scheme inspired by the the technique of "trunk reservation" used in PSTN to allocate circuits to secondary paths in case the direct paths were experiencing overload. [30]

Each of above-mentioned architectures have its counterpart that make no use of path diversity. It appears difficult to unequivocally select one approach. For instance, the question of the deployment of a certain congestion control algorithm is still hotly debated (see the newsletter [40]).

The use of experimental protocols (e.g. cubic in Linux) scares, as **FD** architectures assume a common form of cooperation between users and anarchy might be costly, whether degenerates in congestion collapse, or being less negative, in unfairness in sharing resources.

Manifest truth is that **FD** are very simple and do not require any sort of parametrisation, especially at nodes, and extensions that make use of path diversity are easy to deploy as source routing is made available. However, the main concern that obstructs the deployment of source routing is security.

QD enhances **FD** and make it more efficient and stable. XCP [17] is an example of protocols of this kind while, in the context of intra-domain TE, TEXCP [16] and TRUMP [12] belong to this class.

5.2 Flow aware architecture

5.2.1 Per-flow fair queueing

The benefit of per-flow fair queueing has long been recognised [7, 28]. Besides, it is robust againts unfair use of resources because of non standard conformant use of network transport protocols resulting from bad implementations (rare), absence of common agreement (the case of cubic in Linux [40]) as well as malicious use that exploits others' weakness (more aggressive congestion controllers for instance). Per-flow fair queueing relieves the network to assume standard conformance of end to end protocols. Assured fairness allows new transport protocols to be introduced without relying on detailed fairness properties of preexisting algorithms.

5.2.2 Overload control

Per-flow fair queueing is feasible and scalable in presence of overload control [25, 26]. When demand exceeds capacity the scheduler assures equal performance degradation to all flows. However, arising congestion at flow level is a transient phenomena, as users would quit the service in crowds bringing back utilisation to normal loads. This is perceived by the user as service break-down.

At present, no overload control is implemented, in any form, within the network. At a certain extent, this is assured within the ISPs backbone, and in part within the access, by over-provisioning link capacities according to an estimated traffic matrix (using netflow or similar tools). Such methodology does not solve local congestion in small periods of time and force users to quit profiting from a service.

In wireless mesh backhaul overprovisioning would not be anymore a feasible solution and multi-path routing might be necessary.

Overload control has to be dynamic and fast reacting to congestion. This can be realised looking for resources from other available ways, for instance the availability multiple routes to join a service.

5.2.3 Per-flow path selection and admission control

A sub optimal approach is to select one single path among many others. In presence of a number of paths to reach a destination, a flow can be deflected to a better route, e.g. with larger fair rate or with minimum cost. Such a greedy scheme is not stable and would led to oscillations if the driving metric is not stable enough. This is the case of fair rate.

5.2.4 Per-flow multiple path routing

The presence of per-flow fair queueing in every link impose per-path fair bandwidth sharing. This means that the utility of a single user is not a function of the total attained rate as he is not allowed to get more bandwith of the fair share along its minimum cost path. Therefore the utility is given by the sum of the utility of each singular path as (10). We know that in general this problem is suboptimal with respect to a coordinated splitter. In this case the optimum attained rate is given by $x_i^d = \partial_{x_i^d} U(q_i^d)^{-1}$ being q_i^d the cost of path i for user d. The split ratio is the inversely proportional to the cost. In a recent paper Key et al. [22] prove that in absence of coordination, the stability region of the number of flow in progress in the network, is reduced. This is shown for a triangle topology and uniform traffic matrix. The flow model assumes that flows arrive according to a stochastic process and leave the system after being served an amount of data which is distributed. A reduced rate region at flow level is a consequence of (10) as, the same rate is obtained at a larger cost. As we show in the example 3.4.1. Notice also that in presence of uniform traffic matrices, the over-head requested by multi-path is not really justified with respect to minimum cost path selection as there is almost no gain, as we have shown in the previous section.

5.3 Multi-path Iterative Routing Traffic Optimizer

In this section we propose one of the main outcome of the paper: Multi-path Iterative Routing Traffic Optimiser (MIRTO), a fully distributed algorithm aimed at obtaining, in a decentralised way, a multi-path optimal strategy proposed in sec.3.5.

It is designed around TCP (AIMD) and it allows coordinated traffic split over multiple paths. MIRTO might likely run on end-hosts and can work in any of the architectures described in sec.5.1.

In order to run MIRTO, hosts should discover available routes and link capacities along them to reach a destination. The way such information is collected is out of the scope of our algorithm but they can for example be discovered through link-state routing protocols. The algorithm works even whether information is incomplete but in a less effective way, as in case paths are partially discovered and path diversity limited.

Flow splitting over multiple paths can be performed in several ways according to the application context. By means of path computation capabilities of MPLS for intra-domain TE, or through middlewares running on proxy nodes active as middle-layer.

Symbol	Meaning
Δ^+	positive step $\in \mathbb{R}^+$
Δ^-	negative step $\in \mathbb{R}^-$
$Q_k^d(t)$	cost of k^{th} path of flow d at time t
RTT_k^d	Round Trip Time of the k^{th} path of flow d

Table 2: Algorithm's notation

Algorithm 1 shows MIRTO pseudo-code for rate control over a given path k for a given flow d. Notation is reported in table 2. Operations are performed every RTT_k^d or in general when new information are available on the state of path k. Path costs are computed according to (14). They only depend on capacity of the link along the route. Indeed, in case of linear cost (3) is equal to $\sum_{ij \in L_k^d} \frac{1}{C_{ij}}$. Link with larger capacity are better ranked regardeless of utilisation. Infinite cost in

(14) just indicates congestion notification à la TCP and the route is marked congested.

$$Q_k^d(t) = \begin{cases} \sum_{ij \in L_k^d} \frac{\Delta^+}{C_{ij}} & \text{if } \forall (i,j) \in L_k^d \;\; \rho_{ij}(t) < C_{ij} \\ \infty & \text{if } \exists (i,j) \in L_k^d \;\; \rho_{ij}(t) \geq C_{ij} \end{cases} \tag{14}$$

Algorithm 1 MIRTO algorithm for a given demand d

for $k \in P^d$ **do**
  compute $Q_k^d(t + RTT_k^d)$
end for
if $Q_k^d(t + RTT_k^d) = \infty \;\; \forall k \in P^d$ **then**
  for $k \in \; P^d$ **do**
    $x_k^d(t + RTT_k^d) \leftarrow x_k^d(t) - x^d(t)\Delta^-$
  end for
else if $p^d(t) \geq x^d(t)$ **and** $Q_k^d(t + RTT_k^d) = \min_k Q_k^d(t + RTT_k^d)$ **then**
  $x_k^d(t + RTT_k^d) \leftarrow x_k^d(t) + \Delta^+$
else if $p^d(t) < x^d(t)$ **and** $Q_k^d(t + RTT_k^d) = \max_k Q_k^d(t + RTT_k^d)$ **then**
  $x_k^d(t + RTT_k^d) \leftarrow x_k^d(t) - \Delta^-$
end if

An increase of Δ^+ is done over the minimum cost path when there is at least one non congested path, and the total flow rate can still get increased if lower than its own peak rate. The rate increase is chosen constant and independent to flow rate not to favour higher rate flows. This increase can be considered as a probe in order to discover the global optimum split. In principle all path might be probed as, when those better ranked are congested, worse ranked routes can be exploited. The importance of probing in such kind of controller to avoid to be trapped in non optima equilibria has been highlighted in [21].

Flow rate is decreased in two ways according to network conditions. Firstly MIRTO reduces sending rate over all paths when they are all congested. The rate decrease is proportional to the total flow rate. This guarantees fairness. Otherwise, small flows or new coming flows starting with lower rates would be disadvantaged with respect to higher flows. Moreover decrement is performed over all paths at the same time in order to allow flow split re-arranging.

In fact, after this decrease, there is newly available bandwidth and flows will grow according to the aforementioned rules. So, if the current split is only locally optimum the controller would move towards a global optimum according to this search strategy. On the other hand, a rate decrease is performed any time the total flow rate is larger than flow peak rate and at least one non-congested path is available. This means that, as a better ranked route is willing to increase its rate, this must be done to the detriment of a worse ranked route, without altering the global rate which is bounded by the exogenous rate. In fact even if the total flow rate is equal to peak rate, the path splitting could be only locally optimum i.e. more expensive. It worth recalling that a more expensive path might also be characterized by larger end to end delays. If more than one maximum/minimum cost paths exist, the decrease/increase is shared among them.

This mechanism allows MIRTO to reach a global optimum in presence of demands with or without exogenous rate as it follows the classical water filling procedure that lays at the base of the max-min fairness criteria.

The selection of the minimum-maximum cost path when performing increase/decrease operations guarantees coordination between different paths of the same flow as long as the network does not impose any additional fairness semantic.

5.4 Stability and optimality

MIRTO can be described through a fluid equation that approximates its behaviour and can be used to prove convergence and stability. We suppose first no network delay and then we generealise to the realistic case.

5.4.1 Absence of network delays

First consider the case the flow has no exogenous rate.

$$\frac{dx_i(t)}{dt} = \Delta^+[1 - q(t)]\kappa_i(t) - \Delta^- q(t) \sum_{k=1}^{N} x_k(t)$$

$\kappa_i(t)$ is the probability that i is the minimum cost path at time t. $q(t)$ is the probability that all path are congested. Therefore at steady state

$$\sum_k x_k(\infty) = \kappa_{\max}(\infty) \frac{1 - q(\infty)}{q(\infty)} \frac{\Delta^+}{\Delta^-}$$

where $\kappa_{max}(\infty) = \max_k \kappa_j(\infty)$. This means traffic is split among minimum cost paths, possibly a single path. All the others are subject to continuous probing such that $x_i(t) \sim \kappa_i(t)$. This feature allows the controller not to be trapped in equilibria points that are not optima. A path is probed as frequent as it is ranked the best among the others. This can be seen from simulations in Section 5.5 in fig. 8. In case the flow is peak rate limited the previous equation can be rewritten as follows.

$$\frac{dx_i^d(t)}{dt} = \{\Delta^+ \kappa_i(t) s^d(t) - \Delta^- r_i(t)[1 - s^d(t)]\}[1 - q^d(t)] + -\Delta^- q^d(t) \sum_k x_k(t)$$

where $s^d(t) = Pr[x^d(t) > p^d]$ and $r_i(t)$ is the probability that i is the more expensive path. Convergence can be discussed as in the previous case when there were no peak rate.

5.4.2 Presence of network delays

In presence of network delays $\kappa(t)$, $q(t)$, $r(t)$ and $s(t)$ are delayed information at the source. A problem of stability in the sense of theory of control arises. We do not provide here rules on how to set Δ^+ and Δ^- in order to keep the system asintotically stable in presence of delays. Using standard techniques as described in [33] this can be easily obtained for simple topologies. Cumbersome calculations, and the use of the generalised Nyquist criterion can be used to prove stability for a general topology.

5.5 A case study

In this section we evaluate the performance of the above mentioned architectures by means of fluid simulations in order to diplay the convergence behaviour of three selected scenarios. Each scenario represents one of the three architectures considered in Section 5.1

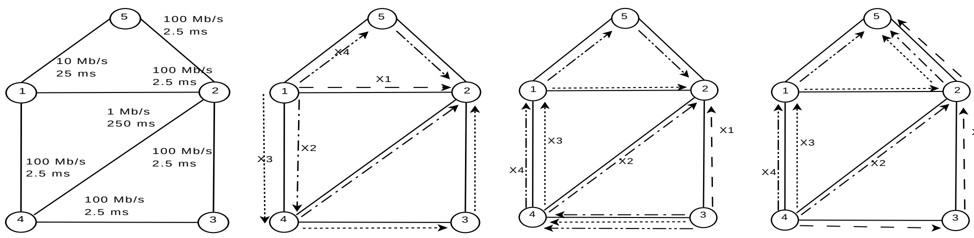

Figure 7: Simple network topology with the set of possible path for each of the three demands.

5.5.1 Simulation setup

The FD and the FA architectures are analyzed by supposing nodes run the MIRTO algorithm. Recall MIRTO has been specifically designed for FD solutions and requires to set rate increase and decrease values. They should allow the algorithm to overcome local optimal split to attain the global optimum and, at the same time, limit traffic fluctuation. The problem of setting increase-decrease values is a well known trade-off of TCP and TCP-like protocols and MIRTO inherits it as well. In our simulations we set them to $\Delta^+ = 0.5Mb/s$ and $\Delta^- = 0.013$.

As concern QD architectures, we implement a modified version of the TRUMP [13] algorithm. It differs from the original one simply because it can work even in presence of flows with a given peak rate. This is achieved by decreasing the rate of a flow over all paths when its total rate overcomes its peak rate. TRUMP requires to set three parameters. A first parameter, called w is a weight to adjust balance between utility and cost function. It is strongly related to topology and link capacities and it tunes the maximum network load. A second one, called β, weighs the impact of a congested link, while a third one, called γ, is the increase/decrease step. For our comparison purposes we set them to $w = 10^{-2}$ to allow flows to fully utilize links, $\beta = 10^{-3}$ to respect link capacities and $\gamma = 10^{-3}$ to limit rate oscillations.

	Time [sec]						
	0-5	5-10	10-18	18-25	25-40	40-60	60-80
X^1	0	0	70	∞	∞	∞	∞
X^2	0	30	30	∞	∞	∞	∞
X^3	50	50	50	50	50	∞	55

Table 3: Per user peak rate evolution over time. Rates values are expressed in Mb/s.

5.5.2 Results

Figure 7 reports the topology used in the simulations we show in this section to show the behaviour of MIRTO. Link capacities and latencies have been selected in order to have path diversity and different response time. There are three flows in the network and we select as source-destination pairs 1-2, 3-2 and 4-5. Flow peak rates change over time as specified in table 3. Note that, ∞ is used to indicate elastic flows. This just means flow peak rate is larger than the available network resources. Every flow can be split over four different paths as shown in Figure.

Figure 8 shows the total flow rates and the flows splitting obtained by running MIRTO over a FD architectures. MIRTO rates are compared to those obtained by running the LP optimizer described in section 4. As expected, total flow rates and rate splits over paths achieved by MIRTO follow those obtained by the LP. Fluctuations occur and are more pronounced when two or more flows are elastic. This is due to the probing nature of the algorithm that allows the global optimum attainment.

Figure 9 shows flow rates obtained by running MIRTO over a FQ architecture and a comparison with the LP. Over this architecture only total flow rates follow the trend of the LP while the flow spitting is quite different from optimum. This confirms what stated in section 5 as FA breaks coordination among flows. In that way, flows cannot achieve rates larger than their fair rate over paths. This leads to a sub-optimal solution even if total flow rates are the same as before. In fact, they have been achieved with a larger network cost. As expected, with a FA architecture rate fluctuations are reduced and the algorithm converges in a shorter time.

Finally, Figure 10 reports performance of the TRUMP algorithm. TRUMP achieves optimal rates and splitting but it takes long time to converge and some long term fluctuations occur. This is a consequence of the set of parameters we used. With different values we would have seen a different behaviours with much faster convergence and no fluctuations. However the optimisation

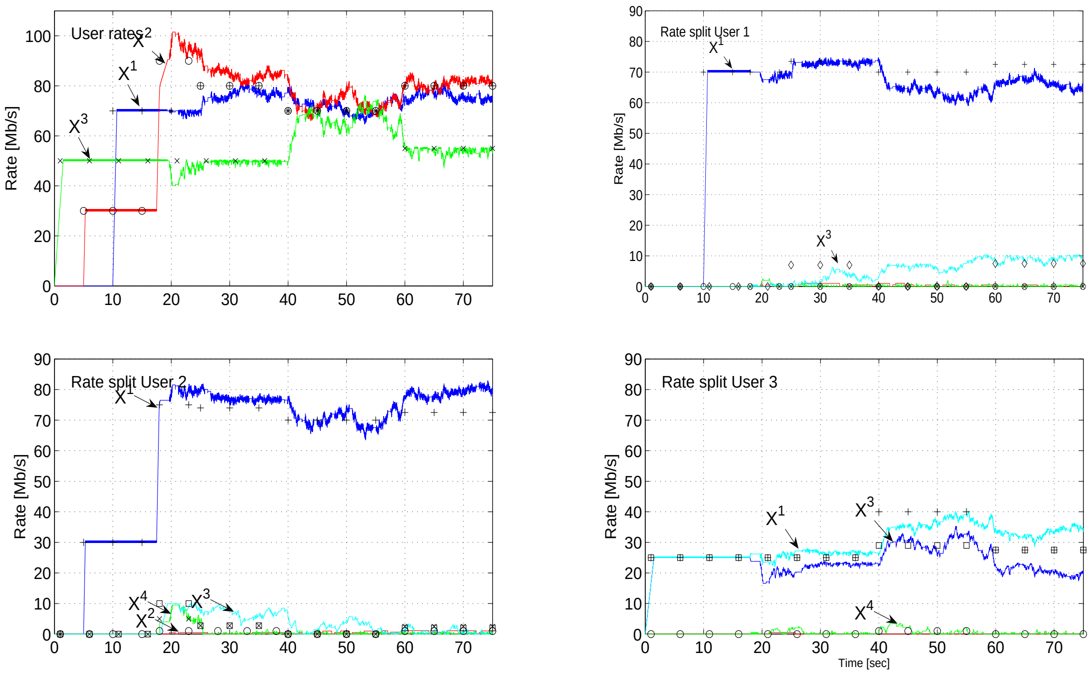

Figure 8: Time evolution of the rate allocation for MIRTO in a FD architecture. Points indicate the optimal allocation.

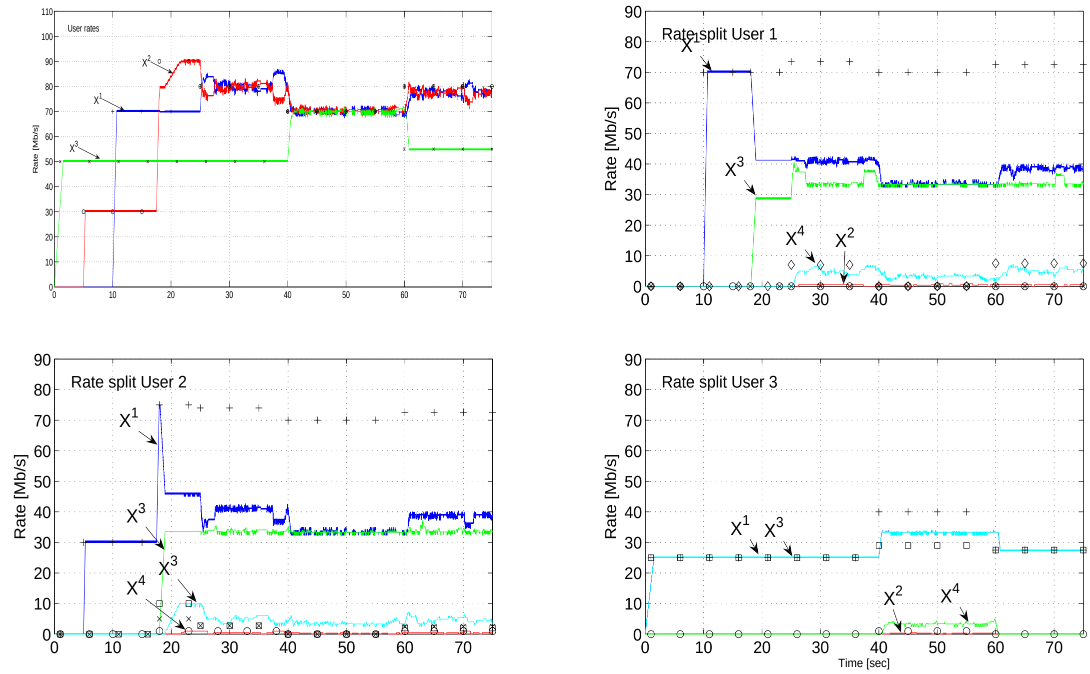

Figure 9: Time evolution of the rate allocation for MIRTO in a FA architecture. Points indicate the optimal allocation.

solution would have been significantly different from that we want to achieve. Actually this is not a drawback of TRUMP as it has been designed in the context of intra-domain TE to limit the link loads within the network, even though authors assume possible to implement such controller at end hosts. Unfortunately TRUMP does not allow to predict at which level of utilisation link can be set. Hence the choice of parameters might be very complex in networks with heterogeneous capacities.

We have tested TRUMP for different setup and different parameters and measured its good properties in term of fast convergence as the authors in [13].

This show the choice of parameters in such algorithms is a key point and could be quite complicated. Moreover it is strongly related to topology and could drive to very different solutions.

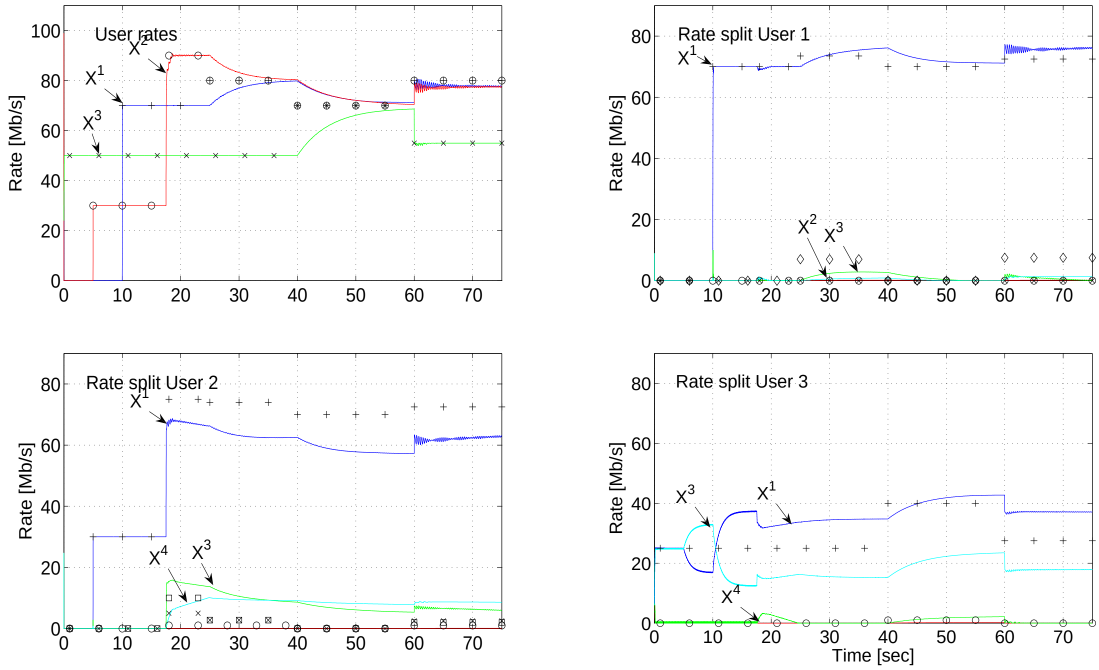

Figure 10: Time evolution of the rate allocation for TRUMP in a QD architecture. Points indicate the optimal allocation.

6 Discussion and Conclusions

The outcomes of this papers are multiple:

- we have studied congestion control and multi-path routing in presence of demands with exogenous peak rates and we have found that multiple routes can be likely exploited by higher rate flows.
- in network scenario that are common in practice, coordinated and uncoordinated multi-path routing perform the same.
- in typical scenarios where coordination outperforms un-coordination, multiple path does not gain much with respect to single path routing.
- furthermore we propose a new multi-path optimal controller called MIRTO. This controller is able to allocate max-min bandwidth among demands exploiting path diversity. The con-

troller has been compared with other solutions and has proved to be easy to be deployed in different architectures.

The above mentioned outcomes allow to positively rethink a flow aware architecture, intrinsically uncoordinated, in presence of multi-path routing. Indeed multi-path controllers as MIRTO can be effectively used in order to exploit path diversity even when a flow aware network imposes fairness at a link base. Task of the controller remains to dynamically define split ratios over the paths as network conditions change.

We suggest the use of multiple paths for a certain class of applications that are naturally robust to variability as adaptive video streaming, P2P file sharing and CDN as they can effectively exploit unused capacity within the network in both Internet backbones and wireless mesh backhauling systems. On the other hand, taking into account the overhead required by multiple routes transmissions, this should not be used for more conversational traffic as voice, web, or short transactions as mail or instant messaging.

References

[1] Abilene Backbone. `http://abilene.internet2.edu/`.

[2] Andersen D.G., Snoeren A.C., Balakrishan H., Best-Path vs. Multi-Path Overlay Routing ACM IMC 2003.

[3] Bertsekas, D. Gafni, E. Gallager, R. Second Derivative Algorithms for Minimum Delay Distributed Routing in Networks, IEEE Transactions on Communications 1984

[4] Bertsekas, D. and Gallager, R., *Data Networks.* Prentice-Hall.1987.

[5] Bertsekas, D. P. (1999). *Nonlinear programming.* Belmont, MA 02178-9998: Athena Scientific. Second edition.

[6] Cantor D. G. and Gerla M., Optimal routing in a packet switched computer network, IEEE Trans. Comput., vol. C-23, pp. 1062-1069, Oct. 1974.

[7] Demers A. , Keshav S. , Shenker S., Analysis and simulation of a fair queueing algorithm, Internetworking: Research and experience, Vol 1, 3-26, 1990. (Also in proceedings of ACM Sigcomm 89).

[8] Fortz, B. Thorup, M. Internet traffic engineering by optimizing OSPF weights, IEEE Infocom 2000

[9] Fortz, B. Thorup, M. Optimizing OSPF/IS-IS weights in a changing world, IEEE JSAC 2002

[10] Gallager, R. *A Minimum Delay Routing Algorithm Using Distributed Computation.* IEEE Transaction on Communications, Vol. 25, No. 1, Jan. 1977.

[11] Han H. , Shakkottai S. , Hollot C.V. , Srikant R. and Towsley D.; Multi-path TCP: a joint congestion control and routing scheme to exploit path diversity in the internet *IEEE/ACM Trans. on Networking* Vol. 14 , Issue 6 (Dec. 2006).

[12] He, J.; Bresler, M.; Chiang, M. and Rexford, J. Towards Robust Multi-Layer Traffic Engineering: Optimization of Congestion Control and Routing IEEE Journal on Selected Areas in Communications, June 2007 Vol. 25, No. 5O

[13] He, J.; Suchara M.; Bresler, M.; Chiang, M. and Rexford, J. Rethinking Internet Traffic Management: From Multiple Decomposition to a Practical Approach, Proceeding of CONEXT 2007.

[14] Low, S.H. Optimization flow control with on-line measurement or multiple paths, in Proceedings of 16th International Teletraffic Congress,1999

[15] Low, S.H. A duality model of TCP and queue management algorithms IEEE/ACM Transactions on Networking, Aug. 2003 Vol. 11, No. 4

[16] Kandula, S.; Katabi, D.; Davie B.; Charnie A. Walking the tightrope: responsive yet stable traffic engineering ACM SIGCOMM 2005

[17] Katabi, D; Handley, M and Rohrs, C.. Congestion control for high bandwidth-delay product networks ACM SIGCOMM 2002.

[18] Kelly, F. *Charging and rate control for elastic traffic.* European Transactions on Telecommunications, volume 8 (1997) pages 33-37.

[19] Kelly, F.; Maulloo, A. and Tan,D. *Rate control in communication networks: shadow prices, proportional fairness and stability.* Journal of the Operational Research Society 49 (1998) 237-252.

[20] Kelly, F., *Fairness and stability of end-to-end congestion control.* European Journal of Control, 9:159–176, 2003.

[21] Kelly, F. and Voice, T. *Stability of end-to-end algorithms for joint routing and rate control.* ACM SIGCOMM Computer Communication Review 35:2 (2005) 5-12.

[22] Key P. , Massoulié L. Fluid models of integrated traffic and multipath routing. Queueing System 53: 65-98, 2006

[23] Key P. , Massoulié L., Towsley D. Combining multipath routing and congestion control for robustness, Proceedings of CISS 2006.

[24] Key P. , Massoulié L., Towsley D., Path selection and multipath congestion control; IEEE Infocom2007.

[25] Kortebi A., Muscariello L., Oueslati S., Roberts J., On the Scalability of Fair Queueing, Proc. of ACM SIGCOMM HotNets III, 2004.

[26] Kortebi A., Muscariello L., Oueslati S., Roberts J, Evaluating the number of active flows in a scheduler realizing fair statistical bandwidth sharing, Proc. of ACM SIGMETRICS 2005.

[27] Mo, J. and Walrand, J., 2000. Fair end-to-end window-based congestion control. *IEEE/ACM Trans. on Networking*, 8(5):556–567.

[28] Nagle J. On Packet Switches with Infinite Storage, RFC 970, IETF, 1985.

[29] Nucci A., Sridharan A., Taft N. , The problem of synthetically generating IP traffic matrices: initial recommendations, ACM SIGCOMM Computer Communication Review, 2005.

[30] Oueslati S. and Roberts J. Comparing Flow-Aware and Flow-Oblivious Adaptive Routing Proceedings of CISS 2006.

[31] Paganini F. Congestion control with adaptive multipath routing based on optimization, Proceedings of CISS 2006.

[32] Radunovic, B.; Gkantsidis, C.; Key, P; Rodriguez,P; Hu, W; An Optimization Framework for Practical Multipath Routing in Wireless Mesh Networks Microsoft Technical Report ,MSR-TR-2007-81, June 2007.

[33] Srikant, R. *The Mathematics of Internet Congestion Control.* Birkhauser, 2004.

[34] Srinivasan V., Chiasserini C. ,Nuggehalli P. and R. Rao, Optimal rate allocation for energy-efficient multipath routing in wireless ad hoc networks, IEEE Transactions on Wireless Communications, vol. 3, no. 3, pp. 891–899, 2005.

[35] Voice, T. *Stability of Multi-Path Dual Congestion Control Algorithms* to appear in IEEE/ACM Transactions on Networking.

[36] Voice, T. A global stability result for primal-dual congestion control algorithms with routing, Computer Communication Review, vol. 34, no. 3, pp. 35–41, 2004.

[37] Wang Z. and Crowcroft J. Analysis of shortest-path routing algorithms in a dynamic network environment ACM SIGCOMM CCR 1992.

[38] Xiaojun Lin and Shroff, N.B. ; Utility maximization for communication networks with multipath routing IEEE Transactions on Automatic Control, May 2006. Vol. 51 No. 5

[39] Zhang-Shen R and McKeown N., Designing a Predictable Internet Backbone Network, HotNets III, San Diego, November 2004.

[40] [e2e] Are we doing sliding window in the Internet?. from the archive of “End-to-End Research Group Charter”. Jan 3 2008.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399